

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2018

Roman Ligocki



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## DETEKCE ODLOŽENÝCH ZAVAZADEL VE VEŘEJNÝCH PROSTORECH

DETECTION OF DEFERRED LUGGAGE IN PUBLIC PLACES

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Roman Ligocki

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Václav Uher

BRNO 2018



# Bakalářská práce

bakalářský studijní obor **Informační bezpečnost**  
Ústav telekomunikací

**Student:** Roman Ligocki

**ID:** 169280

**Ročník:** 3

**Akademický rok:** 2017/18

## NÁZEV TÉMATU:

### Detekce odložených zavazadel ve veřejných prostorech

#### POKYNY PRO VYPRACOVÁNÍ:

Nastudujte si metody pro detekci odložených zavazadel a zpracujte jejich přehled. Na základě získaných poznatků navrhnete algoritmus založený na hlubokém učení. Tento algoritmus následně implementujte a otestujte jeho úspěšnost.

#### DOPORUČENÁ LITERATURA:

[1] LV, Fengjun, et al. Left luggage detection using bayesian inference. In: Proc. of IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance. Citeseer, 2006. p. 83-90.

[2] AUVINET, Edouard, et al. Left-luggage detection using homographies and simple heuristics. In: Proc. 9th IEEE International Workshop on Performance Evaluation in Tracking and Surveillance (PETS'06). Citeseer, 2006. p. 51-58.

**Termín zadání:** 5.2.2018

**Termín odevzdání:** 29.5.2018

**Vedoucí práce:** Ing. Václav Uher

**Konzultant:**

**prof. Ing. Jiří Mišurec, CSc.**  
*předseda oborové rady*

#### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Cílem této práce je vytvořit program pro detekci odložených zavazadel na veřejnosti, které mohou být potenciální hrozbou pro jejich okolí. Součástí práce je popis detekční metody pro hledání objektů v digitálním obrazu. Text dále obsahuje popis použité neuronové sítě YOLO a metod pro porovnávání obrazů. V práci je rovněž popis navrženého algoritmu pro sledování objektů a detekci vlastnictví zavazadla. Výsledkem práce je program, který je schopen detekovat odložené zavazadlo na základě jeho pohybu a vlastnictví.

## **KLÍČOVÁ SLOVA**

Odložené zavazadlo, Detekce objektů, Neuronová síť, Bezpečnost

## **ABSTRACT**

Goal of this thesis is detection of left luggage in public areas, that could be possibly dangerous for their surrounding. Thesis describes detection methods for object detection from digital image. Thesis also contains description of used YOLO neural network and description of methods for image comparison. Text also describes implemented algorithm for object tracking and algorithm for detection of ownership. Result of this thesis is program, able to detect deferred luggage based on it's movement and ownership.

## **KEYWORDS**

Left luggage, Object detection, Neural network, Security

LIGOCKI, Roman. *Detekce odložených zavazadel ve veřejných prostorech*. Brno, 2017, 37 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Václav Uher,

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Detekce odložených zavazadel ve veřejných prostorech“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora(-ky)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Václavu Uhrovi, za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Rovněž bych rád poděkoval svým rodičům, díky kterým jsem měl možnost studovat a kteří byli mou oporou v náročných obdobích studia. Rovněž bych rád poděkoval svému bratrovi ing. Adamovi Ligockému, se kterým jsem konzultoval oblast strojového vidění a neuronových sítí.

Brno .....

.....

podpis autora(-ky)

# OBSAH

<b>1</b>	<b>Úvod</b>	<b>9</b>
<b>2</b>	<b>Metody detekce objektů v obraze</b>	<b>10</b>
2.1	Haarovy příznaky . . . . .	10
2.2	Neuronové sítě . . . . .	11
2.3	Výběr detekční metody . . . . .	12
<b>3</b>	<b>Neuronová síť YOLO</b>	<b>13</b>
3.1	Architektura sítě . . . . .	13
3.2	Ukázka detekčních schopností YOLO . . . . .	13
3.3	Nároky na hardware . . . . .	14
<b>4</b>	<b>Porovnávání obrazu</b>	<b>16</b>
4.1	Střední kvadratická odchylka . . . . .	16
4.2	Strukturální podobnost . . . . .	16
4.3	Výběr metody . . . . .	17
<b>5</b>	<b>Detekce vlastnictví zavazadla</b>	<b>18</b>
5.1	Detekce vlastnictví u člověka . . . . .	18
5.2	Detekce vlastnictví pomocí neuronové sítě . . . . .	18
5.3	Detekce pomocí sledování pohybu osob . . . . .	19
5.3.1	Dynamické borcení časové osy . . . . .	19
5.3.2	Euklidovská vzdálenost . . . . .	20
5.3.3	Výběr metody . . . . .	22
<b>6</b>	<b>Způsoby řešení problému na soutěži PETS 2006</b>	<b>23</b>
6.1	University of Montreal . . . . .	23
6.2	University of Zaragoza . . . . .	24
6.3	National Taiwan University . . . . .	24
<b>7</b>	<b>Výsledné řešení úlohy</b>	<b>25</b>
7.1	Struktura programu . . . . .	25
7.2	Detekce objektů . . . . .	25
7.3	Algoritmus pro sledování objektů . . . . .	26
7.4	Detekce odloženého zavazadla . . . . .	27
7.5	Grafické upozornění obsluhy . . . . .	28

<b>8</b>	<b>Testování detekce odložených zavazadel</b>	<b>30</b>
8.1	Testovací videa . . . . .	30
8.2	Kritéria a způsob hodnocení algoritmu . . . . .	30
8.3	Výsledné hodnocení algoritmu . . . . .	30
8.3.1	Video lehké náročnosti . . . . .	30
8.3.2	Video střední náročnosti . . . . .	31
8.3.3	Video vysoké náročnosti . . . . .	31
8.3.4	Vyhodnocení testu . . . . .	31
8.3.5	Ladění detekčních schopností YOLOv3 . . . . .	31
<b>9</b>	<b>Závěr</b>	<b>34</b>
	<b>Literatura</b>	<b>35</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>37</b>



# SEZNAM OBRÁZKŮ

2.1	Haarovy příznaky . . . . .	10
2.2	Perceptron . . . . .	11
2.3	Plně propojená neuronová síť . . . . .	12
3.1	Detekčních schopností YOLO sítě . . . . .	14
5.1	Vlastnictví zavazadel na základě interakce . . . . .	18
5.2	Příklad nákladové matice v DTW . . . . .	20
5.3	Příklad nákladové matice a výsledné párování . . . . .	21
5.4	Znázornění výpočtu Euklidovské vzdálenosti objektů . . . . .	21
5.5	Porovnání euklidovské vzdálenosti a dynamického borcení času . . . . .	22
6.1	Ukázka homografické transformace a mapy kamerového snímání . . . . .	23
7.1	Schéma implementovaného programu . . . . .	25
7.2	Znázornění principu fungování paměti . . . . .	27
7.3	Sledování a označování objektů . . . . .	27
7.4	Rozdíl mezi ladicím a provozním režimem . . . . .	29
7.5	Znázornění vlastnictví a nahlášení upozornění v režimu ladění. . . . .	29
8.1	Náhled na střeženou oblast ze snímku obtížnosti: lehká . . . . .	33
8.2	Náhled na střeženou oblast ze snímku obtížnosti: střední . . . . .	33
8.3	Náhled na střeženou oblast ze snímku obtížnosti: těžká . . . . .	33

# 1 ÚVOD

V dnešní době se mnoho diskutuje o teroristických útocích, které mohou být provedeny uložením bomby v kufru nebo v tašce a poté odpáleny z velké vzdálenosti. Ohrožena jsou převážně nákupní střediska, stanice metra, letiště nebo vlaková nádraží. Právě obava z případného bombového útoku na veřejnosti vytváří požadavek na způsob, jak tyto nebezpečné předměty detekovat.

K podobnému incidentu došlo na Staroměstském náměstí 2. června 1990. Tehdy neznámý pachatel přišel se sportovní taškou k pomníku Jana Husa. Tašku odložil a odešel. Nikdo nemohl vědět, že v tašce se nacházela železná trubka, naplněná ocelovými střepinami a střelným prachem. Po aktivaci bomby časovým spínačem se střepiny rozlétly mezi osoby v blízkém okolí a způsobily tržná zranění. Právě takovému případu by výsledný program měl být schopen zabránit. Pokud by poplach byl nahlášen s dostatečným předstihem, mohli být lidé na náměstí evakuováni a bomba případně deaktivována policejními složkami.

Pro detekci odložených zavazadel je možné dnes použít různé způsoby, mezi něž patří například ostraha objektu, která opakovanými průchody daného prostoru je schopna vyzorovat podezřelé zavazadlo. Dalším možným způsobem detekce je použití cvičených psů, které jsou schopny na základě dobrého čichu vyhledat zavazadla naplněná výbušnou směsí. Jelikož trénink psů a personál ochranky je finančně náročný, vzniká poptávka po automatizaci tohoto procesu. Tato bakalářská práce se věnuje návrhu programu pro řešení takového problému.

Pro detekci podezřelých zavazadel je potřeba v první řadě detekovat objekty na snímku z bezpečnostní kamery. Proto v první části práce budu rozebírat různé metody detekce objektů z obrazu. Dále bude v rámci práce navržen program, který detekci propojí s algoritmem pro sledování objektů. Výsledkem bude upozornění obsluhy programu o možném nebezpečí.

V teoretické části práce budou popsány metody pro detekci objektů (Kapitola 2), metody pro porovnávání digitálních obrazů (Kapitola 4) a algoritmus pro sledování osob a detekci vlastnictví zavazadla (Kapitola 5). Práce dále obsahuje rozbor navrženého programu a zhodnocení jeho schopnosti detekovat odložené zavazadlo.

## 2 DETEKCE OBJEKTŮ V OBRAZE

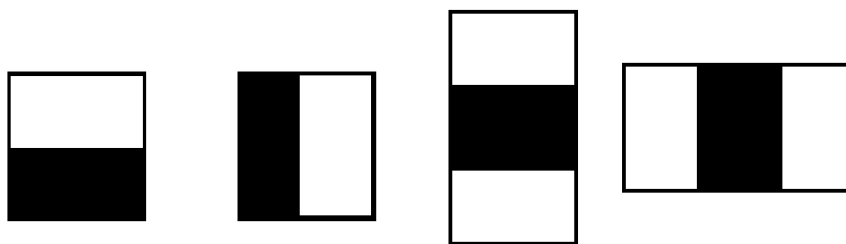
Tato kapitola se bude věnovat popisu nejpoužívanějších metod pro detekci objektů z obrazu. Rovněž bych chtěl shrnout přednosti a nevýhody jednotlivých metod a odůvodnit výběr metody pro účely bakalářské práce.

### 2.1 Haarovy příznaky

Při detekci objektů z obrazu je velice náročné pracovat na úrovni pixelů. Pro vyřešení tohoto problému je lepší vytáhnout z obrazu pouze určité příznaky. Toho si byli vědomi vědci Viola a Jones[4], kteří použili pro detekci obličejů právě Haarovy příznaky. Příznaky se počítají jako součet pixelů ležících v překryté bílé ploše odečtených od součtů pixelů v překryté tmavé ploše. Existuje mnoho příznaků, ale ty nejpoužívanější jsou čárové a hranové. Hranové se skládají ze dvou obdelníků, kde je jeden bílý obdelník a druhý černý obdelník. Čárové se skládají ze tří čtverců, kde dva čtverce jsou bílé a jeden čtverec je černý. Spektrum příznaků je možné rozšířit pokud se pootočí čárové a hranové příznaky například o 45 nebo o 135 stupňů.[3]

Příznaky jsou generovány pro detekční okno. Jeho velikost je možné specifikovat. Při generování příznaků se začíná nejmenším možným příznakem, který se poté posouvá vždy vertikálním nebo horizontálním směrem. Při každém posunu je příznak přiřazen do seznamu. Pokud se příznak posune na okraj detekčního okna, zvětší se jeho velikost. Tento proces se opakuje do chvíle, kdy příznak je větší než detekční okno.[3]

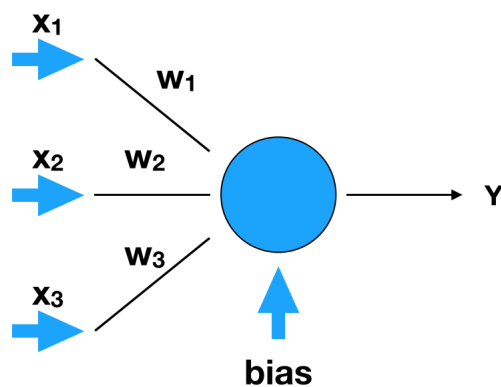
Mezi hlavní přednosti této metody patří nízká výpočetní náročnost při zachování dobré detekční schopnosti. Mezi nevýhody patří nestabilní detekce, kdy na jednom snímku je detekce úspěšná a při minimální změně na druhém snímku je detekce neúspěšná.



Obr. 2.1: Haarovy příznaky

## 2.2 Neuronové sítě

Neuronové sítě jsou v dnešní době velice populární metodou pro tvorbu modelu mezi vstupní vícerozměrnou hodnotou  $x$  a výstupní vícerozměrnou hodnotou  $y$ . Inspirací pro vznik neuronových sítí byla struktura mozkové tkáně vyšších živočichů. Základem neuronových sítí jsou perceptrony, které se snaží napodobovat chování neuronů a tím je nahradit. Perceptron se skládá ze vstupních a výstupních synapsí. Každá vstupní synapse má tzv. váhu, pomocí které se násobí vstupní hodnota na dané synapsi. Výsledek pro každou vstupní synapsi se sečte. K výsledku se ještě připočte bias, neboli vychýlení. Výsledná hodnota se odesílá na výstupní synapsi.[5]

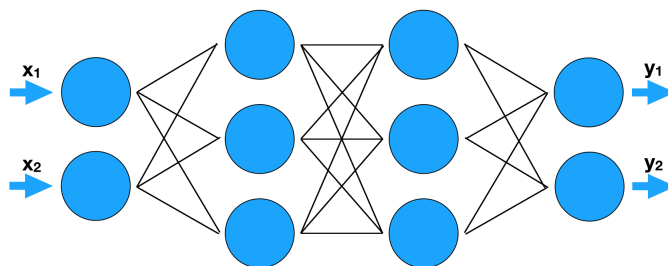


Obr. 2.2: Znázornění perceptronu

Pokud se poskládá více neuronů vedle sebe nebo dokonce vytvoří více vrstev, je možné vytvořit téměř jakýkoliv matematický model, který může reprezentovat například rozpoznávání čísel z obrázků. Čím je neuronová síť hlubší, tím je možné vytvořit sofistikovanější modely. Při vytvoření až příliš hluboké neuronové sítě je možné narazit na problém tzv. přeučení, kdy neuronová síť se výborně naučí model z trénovacích dat, ale není schopná generalizovat tento problém. Poté ověřovací a testovací data mají velice špatné výsledky.[5]

Důležitou součástí neuronových sítí je učení. Jedná se o proces, kdy se vytváří matematický model na základě trénovacích dat. Vždy na vstup neuronové sítě posíláme data a informujeme neuronovou síť o tom, jak má vypadat výsledek. S pomocí algoritmu zpětné propagace se různě nastavují váhy a vychýlení na jednotlivých perceptronech tak, aby při určitých vstupních datech se objevil požadovaný výstup.[5]

Mezi výhody neuronových sítí patří schopnost vytvářet modely, které v mnoha ohledech dosahují kvalit lidského mozku. Z tohoto důvodu je detekce objektů velice přesná a stabilní. Nevýhodou je vysoká výpočetní náročnost, která vyžaduje použití



Obr. 2.3: Znázornění plně propojená neuronová síť

výkonné grafické karty. Bez použití grafické karty je výpočetní náročnost pro běžné procesory příliš vysoká .[5]

## 2.3 Výběr detekční metody

Na základě článku[7] bylo rozhodnuto, že pro bakalářskou práci bude použit detekční algoritmus založen na neuronových sítích. Tato metoda je sice výpočetně náročnější, ale nabízí přesnější a stabilnější detekční schopnosti. Stabilitou detekce je myšleno to, že mezi následnými snímky je detekce téměř shodná s detekcí z předchozího snímku. V počátečních fázích implementace detekčního mechanismu byla odzkoušena metoda haarových vlastností. Tato metoda se ale neosvědčila, protože disponovala velkým množstvím chybných detekcí a docházelo k výpadkům detekce mezi snímky. Při implementaci programu výpadky mezi snímky při detekci způsobovaly ztrátu všech informací o objektu, tedy znemožňovaly následnou detekci opuštěného zavazadla.

## 3 NEURONOVÁ SÍŤ YOLO

Pro účely bakalářské práce byla použita neuronová síť YOLO verze 3 od čtveřice autorů Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. Jedná se o neuronovou síť, kde zkratka označuje anglická slova "You Look Only Once", česky "Podíváš se jen jednou". V rámci bakalářské práce byl použit již předučený model, který je schopen detekovat 60 různých objektů. V tomto případě byla klíčová detekce těchto čtyř objektů: osoba (person), batoh (backpack), kufr (suitcase), taška (handbag). [2]

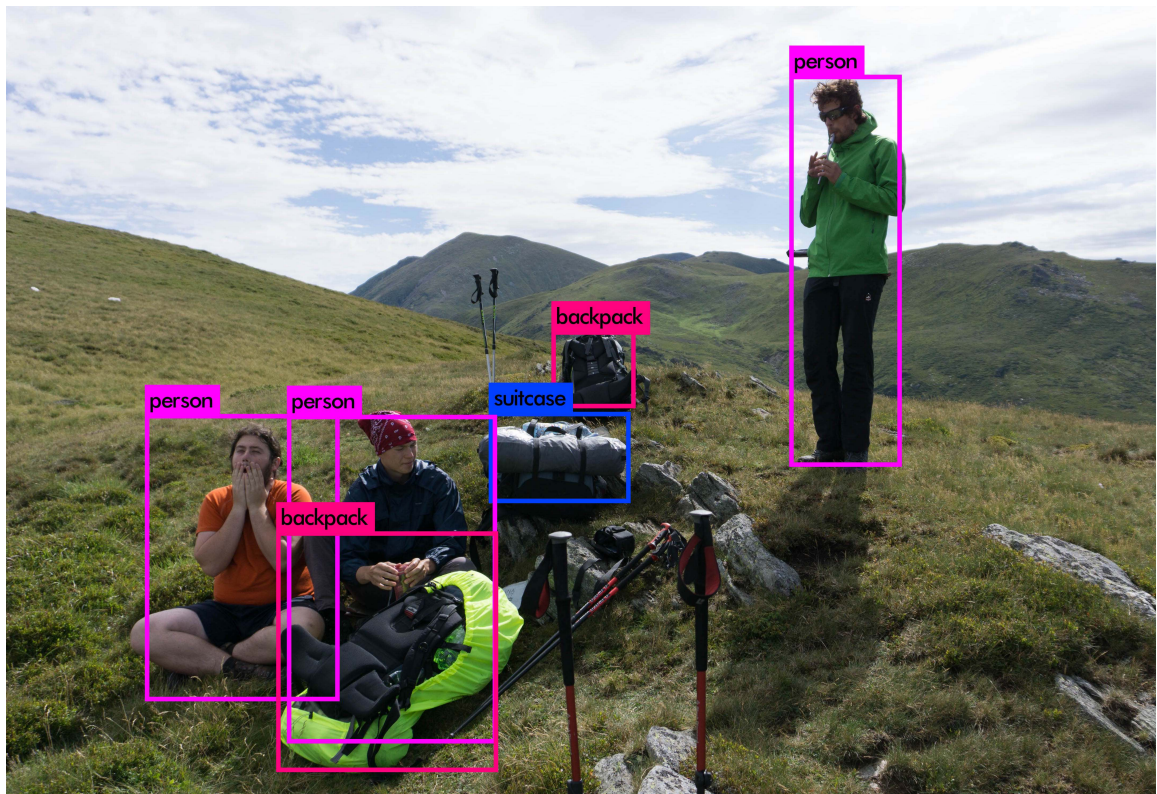
### 3.1 Architektura sítě

Tato neuronová síť je kombinací plně propojené neuronové sítě, nazývané také Fully connected neural network (FCNN) a konvoluční sítě zkráceně Convolutional neural network (CNN). Klíčovou vlastností konvolučních neuronových sítí je vytahování vlastností tzv. feature z obrazu. Příkladem takových vlastností mohou být hrany objektů. Při tvorbě více vrstev konvoluční sítě je možné vytahovat z obrazu více skomplikované vlastnosti jako například lidské oko, nebo tvar postavy. Tyto vlastnosti jsou poté odeslány do plně propojené neuronové sítě, kde podle natrénovaného modelu se vypočítají výstupní hodnoty. Na vstupu neuronové sítě je vždy jeden snímek s pevným rozlišením 608x608 pixelů. Z důvodu, že většina videí má poměr stran 4:3 nebo 16:9, je nutné tyto snímky přepočítat na požadované rozlišení. Tímto přepočtem vzniká deformace obrazu. V tomto případě deformace obrazu neovlivní detekci, protože obrázky použité pro učení, byly rovněž zdeformovány.[2]

### 3.2 Ukázka detekčních schopností YOLO

Na obrázku 3.1 je možné vidět osoby a jejich zavazadla. Z důvodu, že batohy jsou špatně natočeny a je vidět pro batoh málo typických vlastností, bylo pro neuronovou síť velice náročné je detekovat. Výsledná jistota detekce byla 15%. Aby tyto objekty byly detekovány, bylo potřeba snížit prah detekce na 10%. Osoby v tomto případě byly detekovány s jistotou přibližně 70 procent. Jistota v tomto případě je výsledná hodnota od 0 do 1, kde 1 reprezentuje 100 procentní jistotu. Pokud je jistota s jakou neuronová síť objekty detekuje velmi nízká, je velká pravděpodobnost, že výsledná detekce bude chybná. Detekovaný objekt v tomto případě může být špatně klasifikován a nebo dokonce v dané oblasti se žádný objekt nemusí nacházet. Pro otestování náročnosti detekce z pohledu neuronové sítě je možné nahlédnout na pouhý výstřižek daného objektu. Poté člověk nevidí okolí objektu a není schopen určit zda

se jedná o konkrétní typ objektu. V případě že se podíváme na fotografii jako celek, je velice jednoduché určit jednotlivé objekty. Můžeme vidět hory, skupinu turistů, tedy můžeme očekávat, že turisté budou mít s sebou zavazadla a tedy objekty vedle nich budou s velkou pravděpodobností jejich batohy. Neuronová síť v tomto případě není schopna hledat korelaci mezi jednotlivými objekty a uvědomit si například, že turista na horách potřebuje batoh.



Obr. 3.1: Ukázka detekčních schopností YOLO sítě

### 3.3 Nároky na hardware

Cílem této práce je zpracovávat v reálném čase obraz z libovolné bezpečnostní kamery. Z důvodu, že neuronové sítě jsou velice náročné na výpočetní výkon, bylo nutné pro výpočty neuronové sítě použít grafickou kartu. Pro účely vývoje programu jsem používal herní grafickou kartu GTX1060 od společnosti Nvidia. Volba výrobce grafických čipů byla zvolena na základě toho, že grafické karty od společnosti Nvidia mají CUDA procesory, které radikálně navýšují výpočetní rychlost v případě neuronových sítí. S její pomocí bylo dosaženo výkonu přibližně 10 snímků za sekundu. V porovnání s jinými metodami pro detekci objektu, neuronové sítě jsou jedny z nejnáročnějších z pohledu požadavků na hardware. Jelikož v dnešní

době výkon grafických karet roste velkou rychlostí, můžeme očekávat, že příští generace grafických čipů budou tuto problematiku řešit mnohonásobně rychleji. Pro dosažení rychlosti práce s obrazem v počtu 30 snímků za sekundu, bude potřeba použít výkonnější grafickou kartu.



## 4 POROVNÁVÁNÍ OBRAZU

V této kapitole budou představeny metody porovnávání obrazů. V rámci práce bylo potřeba najít způsob jak hledat shodu na podobnost snímků. Představeny budou dva vzorce: střední kvadratická odchylka a strukturální podobnost.

### 4.1 Střední kvadratická odchylka

Sřední kvadratická odchylka, zkráceně MSE je matematický vzorec, pomocí kterého je možné vypočítat rozdíl mezi jednotlivými obrazy. Výpočet se provádí tak, že se sčítá celková odchylka každého pixelu. Každý pixel se skládá ze tří barev. Výpočtená chyba se vydělí celkovým počtem pixelů. Pro výpočet je nutné, aby porovnávané snímky měly stejné rozlišení. Tato metoda počítá rozdíl na úrovni pixelů. U této metody vzniká nebezpečí, že při rychlém pohybu osoby budou změny obrazu mezi snímky příliš velké a výsledný výpočet rozdílu bude příliš velký. Při návrhu programu byl zvolen právě tento výpočet. Ve větším počtu případu vykazoval lepší výsledky v podobě schopnosti hledat podobnost mezi stejnými snímky a naopak. Jeho hlavní výhodou je nízká výpočetní náročnost. [6]

$$MSE = \frac{1}{MN} \sum_{m=1}^{i=0} \sum_{n=1}^{i=0} [I(i, j) - K(i, j)]^2, \quad (4.1)$$

kde  $M$  a  $N$  jsou počty pixelu na řádku a v sloupci.  $I$  a  $K$  jsou hodnoty pixelů na prvním a druhém snímku.

### 4.2 Strukturální podobnost

Strukturální podobnost anglicky Structural Similarity (SSIM) vyjadřuje rozdíl obrazu z pohledu luminance, kontrastu a struktury. Rozdíl snímků se vyjadřuje v rozsahu -1 až 1, kde číslo 1 označuje absolutní podobnost. Tato metoda je již implementována v jazyce python v knihovně scikit-image. Tato metoda je výpočetně mnohem náročnější, než výpočet střední kvadratické odchylky. Při testování tato metoda vykazovala velké nedostatky v podobě velkých výsledných chyb, i při minimální změně obrazu. To způsobovalo špatné fungování implementované paměti v praktické části. [6]

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (4.2)$$

kde  $C_i = (K_i, L)^2$ ,  $L$  vyjadřuje dynamický rozsah hodnot pixelů (pro 8 bitů na kanál tedy 255) a  $K \ll 1$  jsou malé konstanty. Dále  $\mu$  je vážený průměr,  $\sigma$  váženou variací.

### 4.3 Výběr metody

Dle článku[6] je SSIM metoda přesnější. Při návrhu programu bylo zjištěno, že MSE metoda je pro fungování implementované paměti vhodnější. Při ověřování schopnosti sledovat objekt metodou SSIM bylo potřeba snížit hladinu minimální požadované podobnosti natolik, že sledovací algoritmus si začal plést objekty mezi sebou. Pokud byla hladina požadované podobnosti vyšší, objekty se označily novým jedinečným řetězcem, tedy byly brány jako neznámé objekty a tímto bylo přerušeno sledování objektu. Při výpočtu podobnosti pomocí MSE se tato chyba neobjevila a tedy bylo uznáno, že metoda byla pro návrh programu vhodnější.

## 5 DETEKCE VLASTNICTVÍ ZAVAZADLA

V rámci návrhu programu je nejdůležitější nahlásit podezřelé zavazadlo tak, aby nedocházelo ke zbytečným chybným hlášením, které mohou způsobit ignoraci případných hlášení ze strany obsluhy bezpečnostního systému. V této kapitole bude popsáno jak byla detekce navržena a implementována do programu.

### 5.1 Detekce vlastnictví u člověka

Pokud se zamyslíme, jak může lidský mozek detekovat vlastníka zavazadla, tak dojdeme k závěru, že vlastnictví můžeme zjistit na základě interakce vlastníka se zavazadlem a nebo na základě podobnosti pohybu zavazadla a vlastníka. Lidský mozek využívá obou těchto metod zároveň. Schopnost zjistit vlasnictví je možné v takovém případě ve velmi krátkém čase. V rámci této práce byla snaha o napodobení těchto metod, tak aby výsledná detekce byla rychlá a výpočetně minimálně náročná. Pokud se čtenář podívá na snímek 5.1, může okamžitě detekovat vlastníky jednotlivých zavazadel. V tomto případě se jedná o kufřík muže a cestovní kufr ženy.



Obr. 5.1: Vlastnictví zavazadel na základě interakce

### 5.2 Detekce vlastnictví pomocí neuronové sítě

Detekce interakce zavazadla a člověka z obrazu je složitý problém. Důvodem je mnoho způsobů jak vlastník může držet a manipulovat se svým zavazadlem. Kabelka nebo batoh je většinou držen na ramenech nebo na zádech. Kufr je držen rukou za vytahovací mádlo. Bylo by nutné natrénovat neuronovou síť na detekci těchto způsobů držení zavazadla. Tento detail je ale příliš drobný a trénování této detekce by bylo příliš obtížné. Dalším problémem je omezené rozlišení obrazu v neuronové

síti. Po změně rozlišení obrazu, tak aby vyhovovala vstupní matici neuronové sítě, dojde ke znehodnocení obrazu natolik, že není téměř možné tyto interakce mezi člověkem a zavazadlem detekovat. Situace je ještě horší pokud člověk se zavazadlem je velmi vzdálen od kamery zachytávající obraz.

## 5.3 Detekce pomocí sledování pohybu osob

Druhá možnost, tedy sledování detekce vlastnictví na základě pohybu, je pro strojové zpracování daleko přívětivější. Pokud pro každý snímek uložíme všem objektům na snímku jejich pozice do paměti, je možné vytvořit z těchto pohybů křivku. Na základě podobnosti křivek a jejich vzdálenosti je pak možné provést detekci vlastnictví. Během návrhu programu bylo prohledáno velké množství knihoven pro účely hledání podobnosti v křivce. Nakonec došlo k popsání dvou metod, které se běžně používají pro hledání podobnosti mezi křivkami. Obě tyto metody byly implementovány a otestovány. V podkapitolách jsou podrobněji popsány a na základě jejich vlastností je jedna z metod vybrána pro finální podobu programu.

### 5.3.1 Dynamické borcení časové osy

Dynamické borcení časové osy (ang. Dynamic time wrapping) je algoritmus, který se často využívá k porovnávání zvukových signálů. Při porovnávání dvou křivek, které jsou složeny z mnoha vektorů o rozměru  $X$  a  $Y$  je potřeba použít metodu, která umožňuje hledat podobnost i při časovém posunu křivek nebo dokonce při různé časové délce záznamů. Tato metoda je schopna eliminovat případné časové rozdíly a posuny křivek. Pokud je tedy zavazadlo kousek za svým majitelem, ale jeho křivka pohybu je podobná, výsledná podobnost při hledání podobnosti pomocí DTW bude větší než v případě výpočtu podobnosti pomocí Euklidovské vzdálenosti.

Pro výpočet dynamického borcení časové osy je nutné vytvořit nákladovou matici neboli také matici kumulativní vzdálenosti, přes kterou je potřeba najít cestu s minimálním součtem procházejících polí. Na základě této matice se provádí párování jednotlivých bodů. Při této metodě dochází k párování bodů 1:N, tedy jeden bod může být zpárován s více body druhé křivky. První řádek a sloupec je definován následovně:

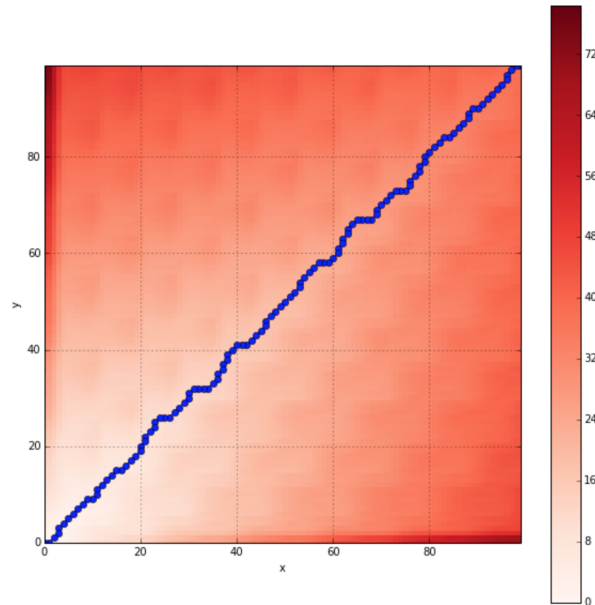
$$D(i, 0) = \sum_{k=0}^i c(r_k, q_0) \text{ pro } i \in [0 : N - 1], \quad (5.1)$$

$$D(0, j) = \sum_{k=0}^j c(r_0, q_k) \text{ pro } j \in [0 : M - 1], \quad (5.2)$$

Zbývající prvky matice se počítají následovně:

$$D(i, j) = \min D(i-1, j), D(i, j-1), D(i-1, j-1) + c(r_i, q_j), \quad (5.3)$$

kde  $i$  a  $j$  je počet sloupců a řádků.  $r_i$  je konkrétní řádek.  $q_j$  je konkrétní sloupec.

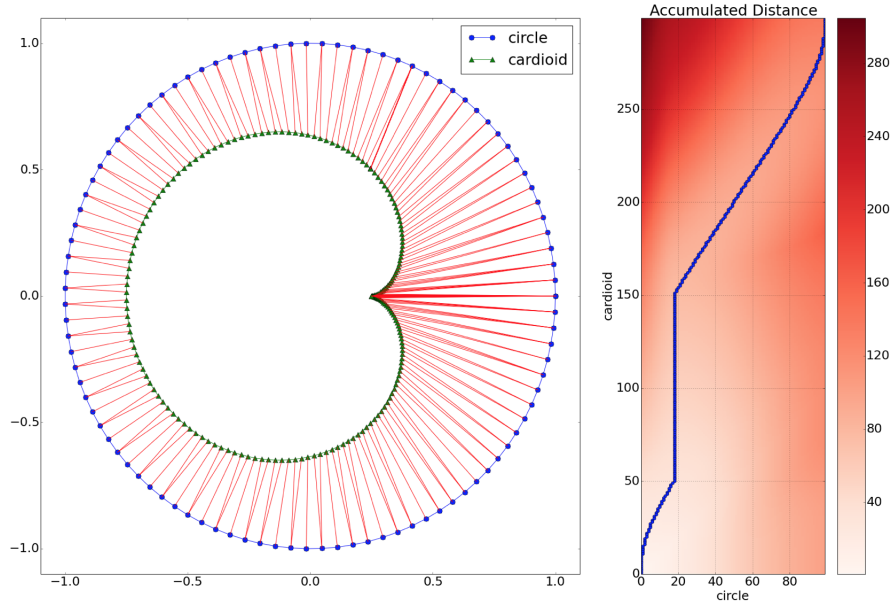


Obr. 5.2: Příklad nákladové matice [13]

Po vytvoření nákladové matice je nutné najít optimální cestu z pozice  $[0,0]$  do pozice  $[X_{\max}, Y_{\max}]$ . Optimum je v tomto případě minimální součet hodnot procházených prvků. Na základě vytvořené cesty dochází k párování bodů, mezi kterými se spočítá euklidovské vzdálenosti. Ty jsou použita jako měřítko podobnosti. Párování probíhá na základě tvaru cesty. Přechod z jednoho prvku matice na druhý prvek matice po ose X vytvoří spojení dvou bodů první křivky s jedním bodem druhé křivky. Pokud by cesta pokračovala dál po ose X, došlo by k vytvoření dalších bodů s stále jedním bodem druhé křivky. Při posunu po ose Y dojde k přesunu na další bod druhé křivky a k němu se přiřadí bod z první křivky. [11]

### 5.3.2 Euklidovská vzdálenost

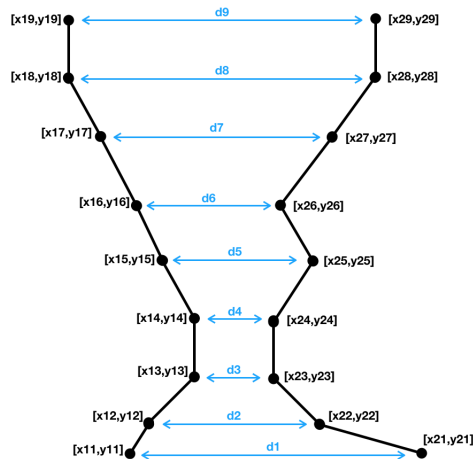
Při návrhu programu došlo k vlastní implementaci funkce pro výpočet euklidovské vzdálenosti mezi všemi body křivky. V tomto případě se jedná o výpočet celkového počtu pixelů, které se nachází mezi dvěma body. Proto, aby výpočet bylo možné provést, je potřeba na vstupu funkce mít dvě křivky se stejným počtem bodů. Tím je možné vytvořit páry a mezi nimi spočítat vzdálenost v pixelech. Tyto vzdálenosti



Obr. 5.3: Příklad nákladové matice a výsledné párování [13]

jsou následně sečteny. K výpočtu je použita Pitagorova věta, která umožňuje vypočítat vzdálenost mezi dvěma body na základě znalostí pozic dvou bodů na ose X a Y. [12]

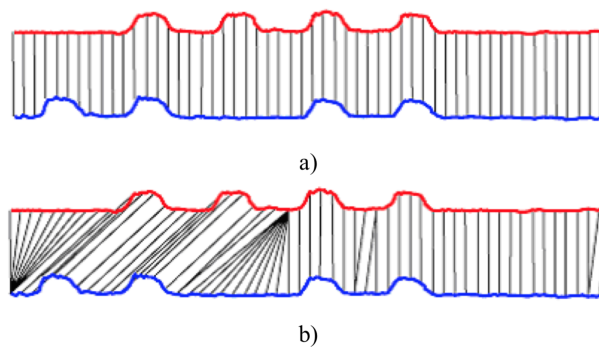
$$s(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (5.4)$$



Obr. 5.4: Znazornění výpočtu Euklidovské vzdálenosti objektů

### 5.3.3 Výběr metody

Při implementaci bylo nutné vybrat jednu z výše popsaných metod pro porovnání křivek. Z důvodu, že nebyl nalezen objektivní způsob posouzení schopností jednotlivých metod. Nakonec bylo rozhodnuto na základě výsledků detekce vlastnictví z referenčních záběrů. Z toho důvodu při implementaci došlo k nainstalování a otestování obou metod. Rozhodnuto bylo na základě jejich schopností detekovat vlastnictví zavazadla dle pohybu osoby a zavazadla. Při otestování obou metod bylo rozhodnuto pro použití metody Euklidovské vzdálenosti. Tato metoda vykazovala lepší výsledky. Projevovala se menším počtem falešně nahlášených nebo chybně nahlášených vlastnictví.



Obr. 5.5: a) Euklidovská vzdálenost b) Dynamické borcení času [11]

## 6 ZPŮSOBY ŘEŠENÍ PROBLÉMU NA SOU- TĚŽI PETS 2006

V této kapitole bych rád provedl rozbor přístupů k problematice týmů z jiných univerzit. Účelem této kapitoly je především vytvořit širší přehled nad přístupy jiných projektů a tím vytvořit lepší implementaci programu.

### 6.1 University of Montreal

Tento team svůj program navrhoval pro prostředí s více kamerami, které střeží určitý prostor. Pro každou kameru následně provedou homografickou transformaci. Tím docílí vytvoření snímku půdorysu snímaného prostoru. Tyto půdorysy se následně proloží a vznikne mapa snímané plochy (Náhled na obrázku 6.1).



Obr. 6.1: Ukázka homografické transformace a mapy kamerového snímání[9]

Pro detekci osob a předmětů se používá detekce siluet postav, ze kterých se následně vytvoří "stíny". Stíny objektů z jednotlivých kamer se následně proloží a pokud objekt má minimálně 3 siluety je nahlášen jako objekt. Rovněž proběhne klasifikace objektu. Sledování objektu je provedeno na základě prokládání objektů mezi snímky. Objekty s největším proložením pixelů je považován za stejný objekt. V rámci algoritmu provádějí spojování a větvení objektů. Pokud se začnou objekty navzájem překrývat, spojí se do jednoho objektu a označí se názvem většího objektu. Po ukončení překrývání se provede rozpojení a objekty se stanou samostatné. Následně každý je nazván stejně jako při spojení. Tímto způsobem se následně detekuje pachatel. Nepohybující se objekt je detekován, pokud za dobu 3 sekund nezmění svou pozici o více jak 30 cm. Upozornění na opuštěné zavazadlo je nahlášeno, pokud



zavazadlo se nepohybuje a jeho vlastník je vzdálen více jak 300 cm od zavazadla. Poplach je ohlášen pokud po třicetí sekundách je vlastník stále vzdálen více jak 300 cm. [9]

## 6.2 University of Zaragoza

Tým rozdělil svou práci na dvě části. První část se věnuje detekci odloženého zavazadla a pachatele. V druhé části je popsána problematika sledování pachatele pomocí většího počtu kamer. V této podkapitole dojde k popisu první části. Algoritmus v první řadě prováděl dlouhodobou analýzu statického pozadí, které z obrazu odstraňuje. Jakékoliv změny v obraze od původního pozadí je považováno za zavazadlo nebo člověka. Zavazadlo je uznáno zavazadlem, pokud jeho odhadovaná výška je přibližně poloviční člověku a je statické. Stejným způsobem jsou detekovány osoby na obrazu s rozdílem dynamického pohybu. Pokud je nalezeno statické zavazadlo, provádí se mutli kamerová analýza na základě, které se ohlásí informace o odloženém zavazadle a provede se označení podezřelé osoby.[10]

## 6.3 National Taiwan University

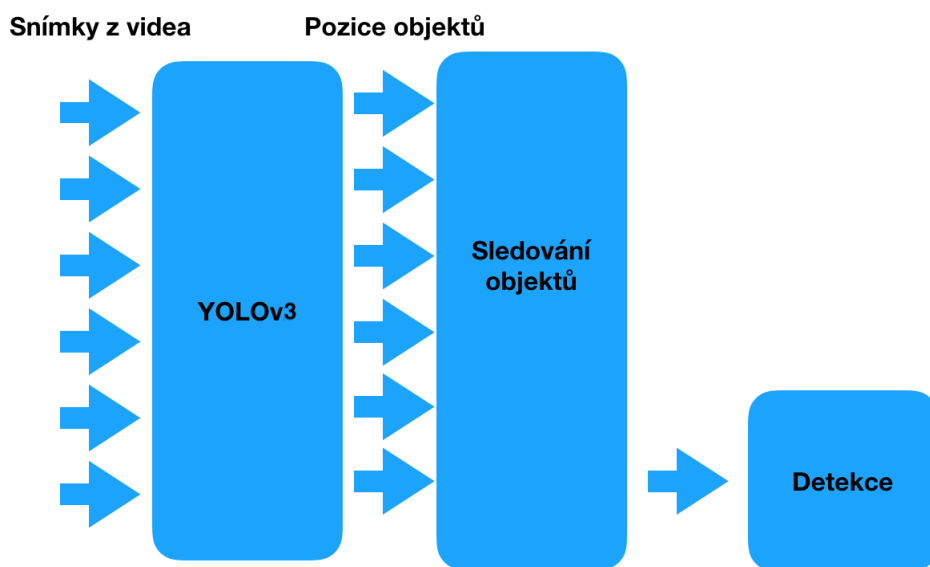
Návrh řešení problematiky tohoto týmu je velice podobný řešení univerzity Zaragoza s rozdílem použití pouze jedné kamery. Na začátku detekce odstraní pozadí vstupních snímků. Obraz bez pozadí projde detekcí osob na základě tvarů lidského těla. Pokud je objekt detekován jako člověk, je tento objekt dočasně ignorován. Zbylé objekty jsou zhlášeny jako zavazadla. V dalším kroku se pro každé zavazadlo hledá vlastník. Hledání vlasníka se provádí pomocí hledání osob v blízkém okruhu zavazadla. Pokud zavazadlo nemá blízko sebe žádnou osobu, dochází k hledání vlasníka na předchozích snímcích. Pokud je nalezen na předchozích snímcích, vlastník je nahlášen jako pachatel a zavazadlo je označeno jako opuštěné.[8]

## 7 VÝSLEDNÉ REŠENÍ ÚLOHY

V této kapitole budou popsány postupy implementace programu navrženého během tvorby bakalářské práce. Jejím cílem je, aby čtenář po přečtení kapitoly byl schopen implementaci napodobit při řešení podobného problému. Popsané jsou jednotlivé postupy a použité balíčky pro programovací jazyk Python.

### 7.1 Struktura programu

Výsledný program se skládá ze čtyř částí: detekce objektů, sledování objektů, detekce odložených zavazadel a grafického upozornění obsluhy. Na obrázku 7.1 je znázorněna struktura programu graficky. Na vstupu programu je video ze kterého se každý snímek odesílá do neuronové sítě YOLO. Ta na svém výstupu má pozice objektů pro každý snímek. Následně se data zpracují a provede se sledování objektů, na základě kterého se provede detekce odloženého zavazadla. Pokud je takové zavazadlo nalezeno, dojde ke grafickému upozornění na objekt obsluze zabezpečovacího systému.



Obr. 7.1: Schéma implementovaného programu

### 7.2 Detekce objektů

Jak již bylo řečeno v teoretické části, pro detekci objektů je použita neuronová síť YOLO. Jelikož celý program byl navrhován v jazyce Python, bylo potřeba tuto neuronovou síť převést do podoby, kterou podporuje jeden z balíčků pro Python zvaný

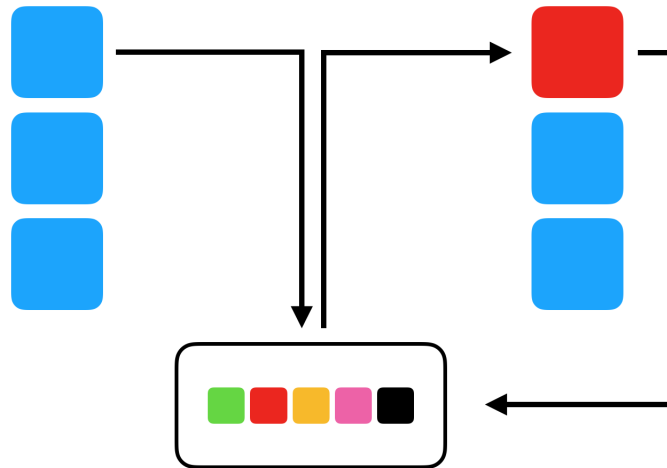
Keras, který vytváří programátorsky přívětivou vrstvu pro tvorbu neuronových sítí. Pod touto vrstvou se nachází balíček tensorflow, který se stará o všechny výpočty na úrovni neuronové sítě. Pro práci s YOLO sítí byl zvolen wrapper keras-yolo3 od autora tanakataiki[14]. Tento wrapper již obsahuje funční neuronovou síť a bylo potřeba pouze přidat ke zdrojovým kódům soubor s předučeným modelem. Při práci s neupraveným wrapperem, se objevil problém, kdy wrapper byl velice pomalý. Důvodem byla deklarace a inicializace všech proměnných pro účely výpočtů neuronové sítě, při detekci objektů na každém snímku videa. Proto bylo potřeba wrapper upravit tak, aby se deklarace a následná inicializace proměnných prováděla pouze jednou při startu programu. Toho bylo docíleno odesláním referencí uložených hodnot do wrapperu.

Jelikož neuronová síť pracuje s každým snímkem zvlášť a výstupem jsou pouze pozice objektů a jejich označování na obraze, bylo potřeba vytvořit algoritmus pro sledování objektů.

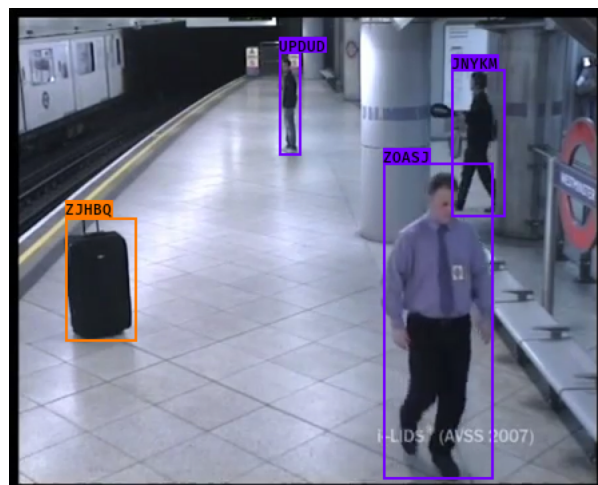
### 7.3 Algoritmus pro sledování objektů

Po tom, co snímek je vytažen z videa a vložen do neuronové sítě, výstupem neuronové sítě jsou pozice a typy objektů, které byly na snímku detekovány. Poté si program vyžádá z databáze objekty z předchozího snímku. Pokud je databáze prázdná, všechny detekované objekty jsou označeny jedinečným řetězcem a uloženy do databáze. Databáze je v programovacím jazyce reprezentována jako proměnná datového typu List. Z důvodu, že se neočekává, velký počet objektů na snímku, nebylo potřeba optimalizovat program lepší datovou strukturou pro urychlení hledání v databázi. V této databázi je každému objektu přiřazen jeho výstřižek, který byl vystřižen ze snímku na základě detekovaných pozic. Při získání pozic a typů objektů z dalšího snímku je pro každý objekt vytvořeno pole v paměti, kde se uloží výsledky porovnání střední kvadratické odchylky pro každý výstřižek z předchozího snímku. Před porovnáním jsou všechny snímky převedeny do rozlišení 100x100. V poli obsahující výsledky porovnání je následně vyhledáno minimum. Na základě indexu nalezeného minima se najde objekt z předchozího snímku. Poté je jedinečný řetězec a aktuální výstřižek překopírován do nového objektu, který se s novými hodnotami uloží do nového pole reprezentujícího aktuální sledované objekty. Objekt s minimální odchylkou výstřižku je po této operaci odstraněn. Pole obsahující aktuální sledované objekty je použito při práci s dalším snímkem. Na obrázku 7.2 je možné vidět názornou ukázkou fungování sledovacího algoritmu. Na snímku je zobrazeno pouze vyhledávání pro červený objekt. Modrou barvou jsou reprezentovány objekty bez přiřazeného jedinečného řetězce. Na základě existence jedinečných řetězců označujících objekty je možné ukládat pozice zavazadel v čase a tímto sledovat, zda se zavazadlo pohybuje

nebo ne. V aktuální verzi u zavazadel program ukládá délku existence objektu na obrazu a změny jeho pozice. Pokud se zavazadlo nepohybuje, tedy jeho pozice na ose x a y se mění v tolerovaných odchylkách, je možné po definovaném čase nahlásit poplach.



Obr. 7.2: Znázornění principu fungování paměti



Obr. 7.3: Sledování a označování objektů

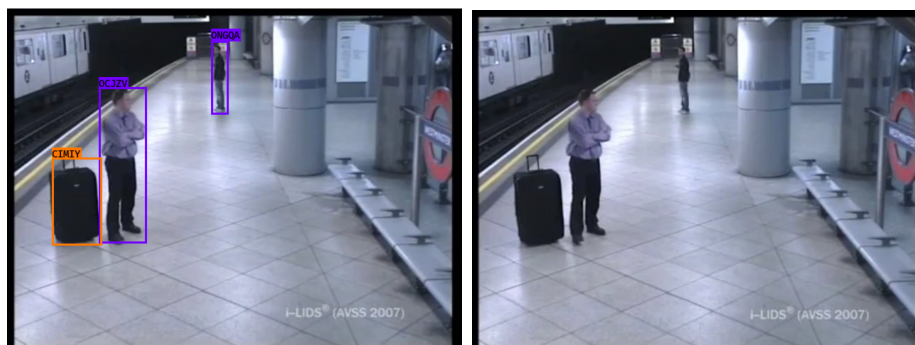
## 7.4 Detekce odloženého zavazadla

Během chodu programu se provádí pro každý snímek detekce vlastnictví zavazadla. K tomu je využita paměť, ve které se ukládají mezi snímky informace o detekovaných objektech. Jednou z položek které se ukládají je posledních n počet pozic v ose X a Y. Pokud zavazadlo má již vygenerovaný n počet pozic v obou osách, prohledá

všechny osoby zda i ony již mají těchto  $n$  počet uložených záznamů. Pokud se provede nalézt taková osoba a zavazadlo není statické, vypočítá se součet vzdáleností mezi jednotlivými body. Pokud je trajektorie pohybu podobná, součet bude větší než, když se budou objekty pohybovat blízko sebe z velmi podobnou trajektorií. V programu je na základě počtu ukládaných pozic vypočítán maximální součet vzdáleností. Pokud součet vzdáleností mezi osobou a zavazadlem přesahuje vypočítané maximum, v ten okamžik dojde ke změně vlastníka zavazadla na prázdný řetězec. Zavazadlo v ten okamžik nemá vlastníka. Po pevně stanoveném počtu snímků se uloží mezi zavazadla, na která je potřeba upozornit obsluhu. Pokud součet nepřesahuje maximální stanovenou hodnotu, uloží se tato chyba do pole, ze kterého se vybere osoba s nejmenší chybou. Poté se uloží do zavazadla jméno jeho vlastníka. V ladícím režimu se rovněž vytvoří grafické znázornění vlastnictví zavazadla zelenou čarou, která je vykreslena mezi středem vlastníka a středem zavazadla.

## 7.5 Grafické upozornění obsluhy

Pokud se zavazadlo zařadí mezi zavazadla na které je potřeba upozornit, provede se oramování daného objektu červenou barvou. Pro účely možného skutečného nasazení produktu byla do programu implementována proměnná, která zajišťuje možnost zapnutí a vypnutí režimu ladění. Pokud je ladění zapnuto, všechny detekované objekty a detekovaná vlastnictví jsou graficky zobrazeny na snímku. Pokud je ladění vypnuto, všechny detekce a případná vlastnictví, kromě poplachu podezřelého zavazadla nejsou zobrazeny na snímku.



Obr. 7.4: Rozdíl mezi ladicím a provozním režimem



Obr. 7.5: Znázornění vlastnictví a nahlášení upozornění v režimu ladění.

## 8 TESTOVÁNÍ DETEKCE ODLOŽENÝCH ZAVAZADEL

### 8.1 Testovací videa

Pro ověřování schopností navrženého programu detekovat odložená zavazadla byla použita videa ze soutěže i-LIDS z roku 2007. V této soutěži měli účastníci vytvořit algoritmus pro detekci odložených zavazadel a zaparkovaných vozidel. Po skončení soutěže došlo k opublikování použitých videí pro akademické účely. Videa jsou celkově tři s tím, že každé video má stanovenou obtížnost. Obtížnosti detekce na video záznamech jsou: lehké, střední a těžké.

### 8.2 Kritéria a způsob hodnocení algoritmu

Protože chybí popis hodnotících kritérií ze soutěže, došlo ke stanovení vlastních kritérií kterými jsou: schopnost detekovat odložená (podezřelá) zavazadla a neohlásit výskyty zavazadla s vlastníkem. Pokud algoritmus bude schopen detekovat zavazadla na všech tří videích a tím upozornit obsluhu, můžeme algoritmus považovat za dostatečně robustní. Současně nesmí docházet k chybně nahlášeným upozorněním, které by snížily důvěru obsluhy ve schopnosti detektoru.

### 8.3 Výsledné hodnocení algoritmu

Pro ověření schopnosti algoritmu detekovat odložené zavazadlo bylo nutné navržený algoritmus spustit na testovacích záznamech. Průběh algoritmu při zpracování jednotlivých záznamů je následně podrobně popsán. Rovněž jsou popsány problémy, které se během vývoje algoritmu objevily.

#### 8.3.1 Video lehké náročnosti

Záznam má délku 3:38. V čase 1:10 se objevuje muž se zavazadlem, přesouvá se do blízkosti kamery a pokládá cestovní tašku. Zavazadlo i muž jsou velmi dobře viditelní. Program vytváří vazbu mezi mužem a cestovní taškou. Po 45 sekundách se muž začíná vzdalovat od zavazadla. Po dalších 15 sekundách je přerušena vlastnická vazba a zavazadlo je označeno za podezřelé. V tomto případě se žádná osoba nepohybovala ani před zavazadlem ani před vlastníkem zavazadla. Je důležité zmínit, že oblast v které se detekce prováděla, nebyla silně zaplněná ostatními lidmi. Detekci na tomto video záznamu je možné považovat za úspěšnou.

### 8.3.2 Video střední náročnosti

Záznam má délku 3:13. V čase 0:50 se do hlídaného prostoru vstupuje žena s velkou sportovní taškou. Taška se pohybuje již ve větší vzdálenosti a její detekce je tedy nestabilní. Po dosažení cíleného místa žena pokládá tašku. Po padesáti sekundách odchází od zavazadla mimo střeženou oblast. Následně dochází k zhoršení detekční schopnosti na úroveň, kdy je zavazadlo detekováno pouze pokud v jeho blízkosti se nachází jiná osoba. Pokud žádná z osob není v blízkosti zavazadla nedochází k jeho detekci. Následně ani nedochází k udržení instance zavazadla v paměti programu a tedy není možné nahlásit případné upozornění.

### 8.3.3 Video vysoké náročnosti

Záznam má délku 3:32. V čase 0:40 vstupuje do střežené oblasti žena s cestovním kufrem na kolečkách. Po dobu příchodu je zavazadlo detekováno s vysokou stabilitou. Se zvětšující se vzdáleností od kamery dochází k stále častějším výpadkům detekce. Po dosažení cílového místa kufr zastaví vedle sebe. Po celou dobu, kdy se kufr nachází v blízkosti ženy je opětovně stabilita detekce vyžší, ale stále dochází k velmi častým výpadkům. Po přibližně minutě žena nechává zavazadlo na místě a střeženou oblast opouští. Následně klesá stabilita detekce zavazadla natolik, že není možné zhlásit zavazadlo jako opuštěné.

### 8.3.4 Vyhodnocení testu

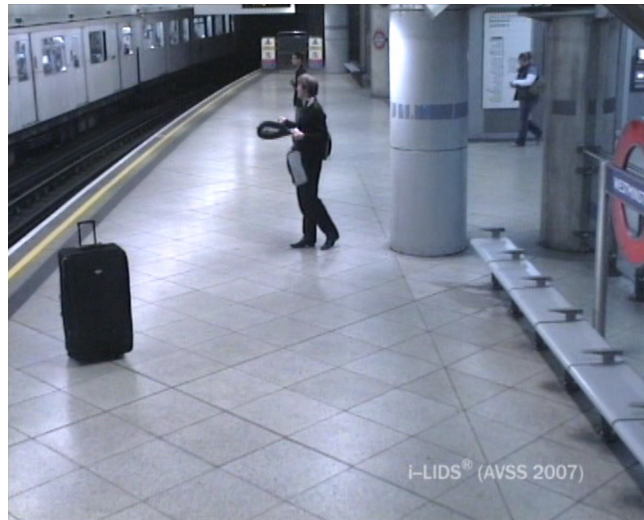
V rámci testu došlo k zjištění, že detekční schopnosti neuronové sítě YOLOv3 jsou nedostačující při detekci velmi malých objektů. Nízká robustnosti YOLOv3 sítě má za následek ztrátu veškerých informací o zavazadle. Následně není možné ani při nízkém výskytu okolních osob detekovat opuštěné zavazadlo. Na základě pozorování z prvního videa je možné říct, že samotná detekce odloženého zavazadla je dostačující pro všechny tři videa a pouze detekce objektů je nedostačující.

### 8.3.5 Ladění detekčních schopností YOLOv3

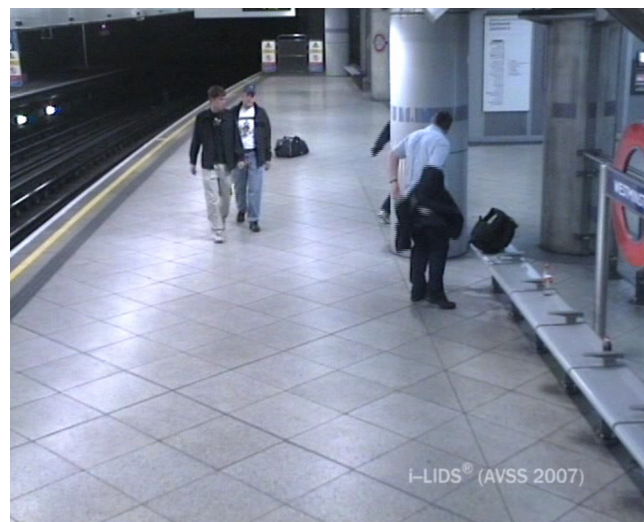
Při návrhu programu byl pro neuronovou síť použit již předtrénovaný model. Ten ale nebyl dostačující, protože nebyl schopen detekovat objekty nacházející se ve velké vzdálenosti. Tyto nedostatečné schopnosti měly za následek, že došlo k pokusu o změnu parametrů neuronové sítě, tak aby vzdálené objekty bylo možné detekovat. Možnosti byly dvě: změnit parametry vstupní matice a nebo přetrénovat model sítě. V prvním případě šlo pouze o změnu rozlišení vstupních snímků v konfiguračním souboru. Nastavit bylo možné jakékoliv násobek čísla 32. Protože rozlišení videa



bylo pouze 720x576, zvýšení rozlišení neuronové sítě z 608x608 na 736x736 nemělo pro detekci velký rozdíl. Zavazadlo na druhém a třetím videu bylo nadále špatně detekovatelné. V tomto případě nebylo potřeba provádět další změny. Druhá možnost byla natrénovat model dle vlastních parametrů. Pro trénování byl použitý trénovací režim v původní implementaci YOLOv3. Jako trénovací data byly použity již klasifikované snímky z datasetu COCO. V konfiguračním souboru před trénováním byla provedena změna v podobě zmenšení počtu detekovatelných typů objektů. Hypotézou bylo, že při učení menšího počtu typu objektů se detekční schopnosti zvýší. Důvodem je, že váhy jednotlivých perceptronů se nebudou měnit vlivem jiných nežadoucích objektů. Po dokončení vlastního trénování nedošlo ke změně detekčních schopností malých objektů.



Obr. 8.1: Náhled na střeženou oblast ze snímku obtížnosti: lehká



Obr. 8.2: Náhled na střeženou oblast ze snímku obtížnosti: střední



Obr. 8.3: Náhled na střeženou oblast ze snímku obtížnosti: těžká

## 9 ZÁVĚR

V dnešní době se stále častěji objevuje problém teroristických útoků, které jsou páčány pomoci výbušnin uložených v kufrech nebo sportovních taškách. Tyto zavazadla jsou následně na dálku odpáleny. Jedinou cestou jak ochránit okolí tohoto zavazadla před výbuchem nálože je její brzké odhalení a zneškodnění. Pro tento účel vznikla má bakalářská práce.

Zadáním práce bylo vytvořit program, který pomocí hlubokého učení bude schopen detekovat odložené zavazadla. Pro správný přístup k problému bylo nutné provést důkladnou rešerši metod pro detekci objektů a porovnávání objektů. Z popsaných metod následně byly vybrány, ty které svými vlastnostmi vyhovovaly nejvíce k účelům návrhu programu. Pro detekci objektů byla použita metoda neuronových sítí, konkrétně se jednalo o síť YOLOv3. Pro následné sledování zavazadel mezi snímky bylo nutné provádět porovnávání výstřížků objektů mezi sebou. Pro tyto porovnání byl nejvhodnější výpočet střední kvadratické odchylky, který porovnával jednotlivé pixely obrazu mezi sebou. Bakalářská práce rovněž obsahuje rozbor řešení problematiky týmů z jiných světových univerzit. Popsané metody jak tento problém řešit byl použitý k rozšíření přehledu a k možné inspiraci jak navržený algoritmus vylepšit.

V další kapitole byl proveden rozbor, jakým způsobem se budou detekovat odložená zavazadla. Výsledkem byla vlastní implementace detekce odloženého zavazadla na základě pohybu zavazadla v čase. Každý objekt, který se nachází na snímku má svou vlastní instanci uloženou v paměti. Do této instance, se ukládají informace o pohybu objektu mezi jednotlivými snímky. Právě tyto informace se používají k detekci vlastníka zavazadla. Každé zavazadlo porovnává svou křivku pohybu s křivkami pohybu osob. Pro porovnávání se používá výpočet euklidovské vzdálenosti. Pokud výsledná podobnost nedosáhne prahové hodnoty je zavazadlu přiřazen vlastník. Pokud se vlastník vzdálí a dojde k překročení prahové hodnoty dojde k zrušení vlastnictví. V tomto případě pokud zavazadlo nenajde jiného vlastníka a zároveň se nepohybuje je zhlášen poplach.

Pro ověření robustnosti celého implementovaného programu byly použity videa ze soutěže i-LIDS z roku 2007. Video byly rozděleny do tří obtížností, kde se měnily vzdálenosti zavazadla od kamery a četnost osob pohybujících se v střežené oblasti. Navržený program obstál bez problému v nejlhčím scénáři, kde zavazadlo se nacházelo v blízkosti kamery a bylo dobře viditelné. V dalších dvou videjích program nebyl schopen detekovat objekty ve velké vzdálenosti. Důvodem bylo nízké rozlišení videa, kdy neuronová síť již nebyla schopna ze snímku zjistit, zda objekt je zavazadlem.

## LITERATURA

- [1] WANG, Z., A.C. BOVIK, H.R. SHEIKH a E.P. SIMONCELLI. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* [online]. 2004, **13**(4), 600-612 [cit. 2017-11-22]. DOI: 10.1109/TIP.2003.819861. ISSN 1057-7149. Dostupné z: <http://ieeexplore.ieee.org/document/1284395/>
- [2] FARHADI, Joseph Redmon Ali. YOLOv3: An Incremental Improvement.
- [3] KLAPAL, Matěj. Detekce rasistických symbolů z obrazu [online]. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií, 2015 [cit. 2017-11-25]. Dostupné z: <http://hdl.handle.net/11012/41355>. Bakalářská práce. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací. Vedoucí práce Lukáš Povoda.
- [4] VIOLA, Paul a Michael JONES. Rapid Object Detection using a Boosted Cas-cade of Simple Features [online]. 2001. [cit. 2017-11-29]. 9 s. Dostupné z: <https://www.cs.cmu.edu/~efro/courses/LBMV07/Papers/viola-cvpr-01.pdf>
- [5] KARÁSEK, Štěpán. Neuronové sítě a genetické algoritmy [online]. Vysoké učení technické v Brně. Fakulta informačních technologií, 2016 [cit. 2017-11-27]. Dostupné z: <http://hdl.handle.net/11012/61862>. Diplomová práce. Vysoké učení technické v Brně. Fakulta informačních technologií. Ústav inteligentních systémů. Vedoucí práce František Zbořil.
- [6] How-To: Python Compare Two Images - PyImageSearch. PyImageSearch - Be awesome at OpenCV, Python, deep learning, and computer vision [online]. Copyright © 2017 PyImageSearch. All Rights Reserved. [cit. 29.11.2017]. Dostupné z: <https://www.pyimagesearch.com/2014/09/15/python-compare-two-images/>
- [7] CENGIL, Emine a Ahmet CINARS. Comparison Of Hog (Histogram of Oriented Gradients) and Haar Cascade Algorithms with a Convolutional Neural Network Based Face Detection Approaches [online]. Computer Engineering Department Firat University, 2017 [cit. 2017-11-29]. Dostupné z: <https://www.ijariit.com/manuscripts/v3i5/V3I5-1189.pdf>
- [8] CHANG, Jing-Ying, Huei-Hung LIAO a Liang-Gee CHEN. Localized Detection of Abandoned Luggage. *EURASIP Journal on Advances in Signal Processing* [online]. 2010, 2010(1), - [cit. 2018-04-10]. DOI: 10.1155/2010/675784. ISSN 1687-6180. Dostupné z: <https://asp-urasipjournals.springeropen.com/articles/10.1155/2010/675784>

- [9] AUVINET, Edouard, et al. Left-luggage detection using homographies and simple heuristics. In: Proc. 9th IEEE International Workshop on Performance Evaluation in Tracking and Surveillance (PETS'06). Citeseer, 2006. p. 51-58.
- [10] MARTÍNEZ-DEL-RINCÓN, Jesús, et al. Automatic left luggage detection and tracking using multi-camera ukf. In: Proc. IEEE 9th IEEE Int. Workshop PETS. 2006. p. 59-66.
- [11] LU, Bo, et al. Constrained selective dynamic time warping of trajectories in three dimensional batch data. *Chemometrics and Intelligent Laboratory Systems*, 2016, 159: 138-150.
- [12] KLÍMEK, Petr. *Shlukovací metody v data miningu*. E+ M Ekonomie a Management, 2008.
- [13] GitHub - ricardodeazambuja/DTW: Dynamic Time Warping in Python / C (using ctypes). The world's leading software development platform · GitHub [online]. Copyright © 2018 [cit. 12.04.2018]. Dostupné z: <https://github.com/ricardodeazambuja/DTW>
- [14] GitHub - tanakataiki/keras-yolo3: A Keras implementation of YOLOv3 (Tensorflow backend). The world's leading software development platform · GitHub [online]. Copyright © 2018 [cit. 18.04.2018]. Dostupné z: <https://github.com/tanakataiki/keras-yolo3>

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

YOLO	You look only once
MSE	Mean squared error
SSIM	Structural Similarity
FCNN	Fully connected neural network
CNN	Convolutional neural network
LSTM	Long-short term memory